



Abstract

The DDR SDRAM controller used in the AMCC PowerPC 440GP and the PowerPC 440GX supports a wide variety of memory architectures, including point-to-point, SO-DIMMs, and DIMMs. In order to support these memory systems, the DDR memory systems read and write timings must be adjustable over a limited range of time values.

These read and write operations are fine-tuned by the digital delay line feature contained in the DDR SDRAM controller. This feature enables the system architect to optimize the memory transfer timings. Without careful consideration, however, the configuration registers can be set up improperly, which could cause the memory system to fail.

This application note assists the memory system architect in understanding key concepts, components, and timing effects when using the digital delay line for the DDR SDRAM memory controller.

Introduction

This application note seeks to clarify one particular area of confusion when using the DDR controller: how to set precise data read and write timing signal delays. The settings are based upon the memory system requirements, timing budget calculations, and resulting delay line timing values returned from the delay line calibration register, SDRAM0_DLYCAL. The returned 8-bit delay line calibration value is the basis for the overall system delay values that are to be used when tuning the memory system's read and write timing.

The calibration value is used when adjusting the timing relationship between the various control and data signals in the interface. The registers that adjust the timings between the control data signals use the count of delay line elements (as opposed to a simple time value). This method enables the DDR system to be fine-tuned for a wide variety memory components, board layouts, and environmental conditions.

The timing relationships between the control, data, and memory data strobe's main memory system clock (MemClkOut0) can be varied. In general, the timing relationships do not need to be manipulated much to make the memory system operate correctly, but adjustments can correct for the various delays associated with the memory system design and layout.

PLB Clock Signal

The processor local bus (PLB) clock found in the PowerPC 440GP/GX is used as the main timing reference for all memory system operations. The operating frequency of the memory system is locked to the PLB frequency, so all memory system timing budget calculations are based upon the PLB clock domain.

The differential signal MemClkOut0 is the main external memory system-synchronizing signal. MemClkOut0 is used to synchronize the DDLs found in the DDR SDRAM components. The DDLs are used to properly launch read data from the memory devices.

Data originates from the PLB bus and flows out to the memory system and its delayed time domain during data write operations. During data read operations, data arrives from the memory system and its time domain. All data driven from the memory system is skewed from the PLB time domain by the delays accumulated from various parts of the physical memory system, including wiring delays, access time delays, and buffer and receiver loading delays. Data from the memory devices, which appears in the memory system time domain, must be synchronized to the PLB time domain. The DDR interface has enough flexibility to enable a wide range of memory systems to operate correctly, but this flexibility also makes it possible to configure the interface incorrectly.

Digital Delay Line and the Calibration Value

The PowerPC 440GP/GX contains a digital delay line that is used to generate fine-tuning timing increments. Essentially, the delay line is built from a series or chain of delay or logic blocks called elements. For the purposes of this application note, the detailed function and contents of the elements are not important, but the time delay per element is.

When the PowerPC 440GP/GX is powered-up, the digital delay line performs a self-calibration function. After the calibration process is completed, the 8-bit register `SDRAM0_DLYCAL[DLCV]` value gives the number of full delay elements timed out in a half-clock cycle of the memory system clock signal, `MemClkOut0`. This is the calibration value. (Another way of viewing this is that it takes a finite amount of time for a signal to propagate through the delay line elements. The 8-bit count found in the register is the count of delay line elements that the signal propagates through in one-half of a memory clock cycle.)¹

Note: The value found in the `SDRAM0_DLYCAL` register should never fill the entire register. Usually, the `DLYCAL` value is only around 30%-40% of the maximum possible count.

The delay line calibrates itself at chip-reset time and loads the calibration value in the `SDRAM0_DLYCAL` register. The digital delay line can also be forced to recalibration itself by setting the `DLCR` bit in the `SDRAM0_DLYCAL` register. Recalibration can compensate for extreme environmental changes (though this should not be necessary) and ensure that the timings for the memory system are always optimal.

Note: When recalibrating the delay line, be sure that all outstanding memory requests are completed. To do this, perform the following steps:

1. Disable the DDR SDRAM controller by writing `SDRAM0_CFG0[DCEN] = 0`.
2. Update the DDR registers.
3. Enable the DDR controller by writing a 1 to `SDRAM0_CFG0[DCEN]` register.

The DDR system will reinitialize the memory system following the standard initialization sequence. Refer to the PowerPC 440GP User's Manual or the PowerPC 440GX User's Manual for more details about the recalibration process.

Note: In order to convert the delay line calibration value into time units, divide the half-`MemClkOut0` cycle time by the number of delay elements found in the calibration register. The result is the time per element.

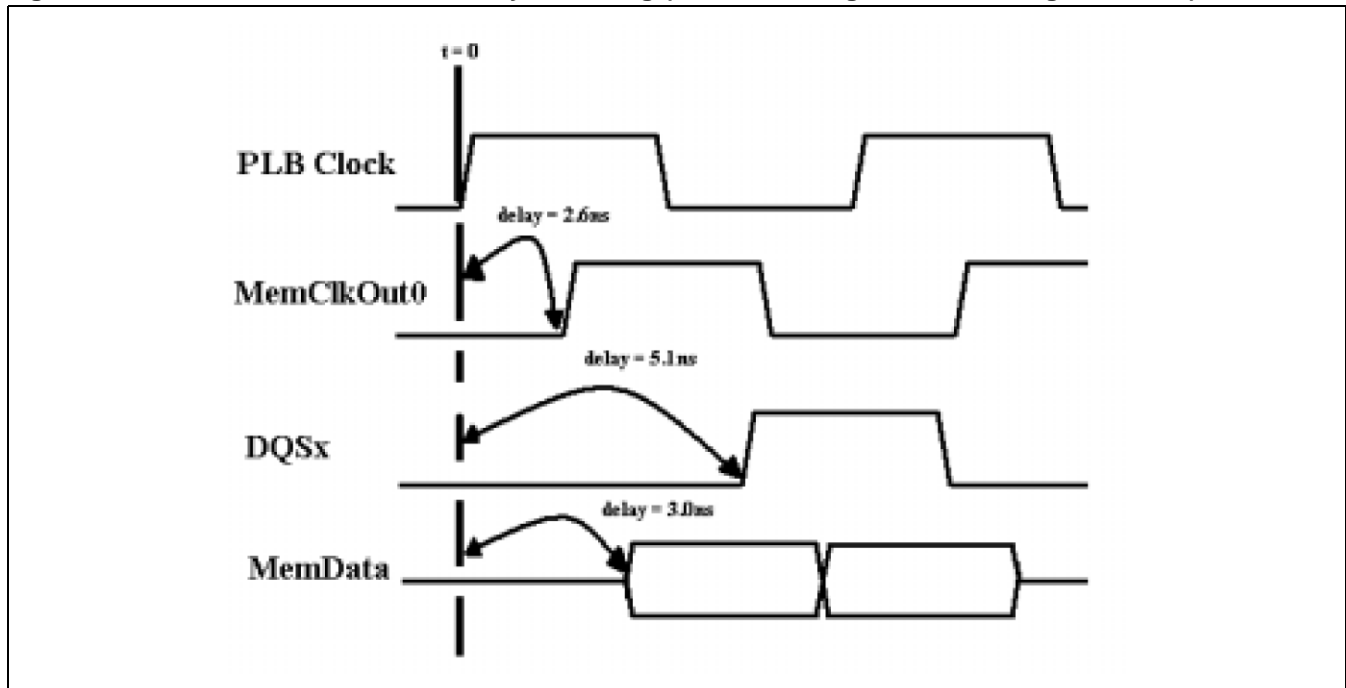
The delay line calibration value is used to optimize the timings for both memory read and write operations. The delay line values can be used to incrementally alter the timing relationship between `MemClkOut0`, data, plus the datas associated strobe signals. `MemClkOut0` can be advanced in large increments of 90°, 180°, and 270°. Advancing `MemClkOut0` by such increments generally corrects the interface timings to the needs of the specification and design. These corrections enable the designer to optimize the relationships between `MemClkOut0` and all data signal groups, data bits, and associated data strobes signals.

The DDR command signal lines operate at the clock rate of the memory system. Data, strobes, and data mask signals operate at twice the memory system's clock rate. For the purpose of this document, the main data signals under consideration are: data bits[0:(31:63)], associated byte lane strobe bits `DQS[0:8]`, and `MemClkOut0`.

Note: All of the data signals associated with a single data strobe signal compose a byte lane. The memory interface found in the PowerPC 440GP/GX is designed to operate with x8 (or by 8) memory devices. This means that for each data byte in the memory system, there is an associated data strobe signal `DQSx`. An x8 device is the only type of DDR memory device that will work with the memory controller.

1. The 8-bit delay line calibration register returns the count of full delay elements required to time out one-half of a DDR `MemClkOut0` cycle (5 ns for 100MHz memory system; 3.75 ns for a 133-MHz memory system; and 3.01 ns for the 166-MHz system). Basically, during a calibration cycle a signal is sent into the delay line. The delay line is then sampled a half-clock cycle later. The progress of the signal through the digital delay line is used to set the number of "taps", "elements", or "stages" in the `SDRAM_DLYCAL` register. This 8-bit register value allows a maximum of 256 taps to be calibrated to the clock cycle. For a 100-MHz memory system, the calibration register offers an ultimate resolution of $5 \text{ ns}/256 = 19.5 \text{ ps}$ per tap or element.

Figure 1: Default DDR SDRAM Write Cycle Timing (Demonstrating the Various Signal Skews)



Data Write Operations

For data writes, the rising edge of the byte lane strobe bit, DQSx, should be edge-aligned with the rising edge of MemClkOut0. Data should appear roughly one quarter-cycle before the leading edge of the strobe signal. (Basically, the data strobe edge is center-aligned with the data.) This follows the DDR JEDEC specification that the strobe signal appears in the middle of data during a data write.

When reviewing the timing information available for the DDR interface used in the PowerPC 440GP/GX, the end user will actually find that MemClkOut0, data bits, and data strobe signals are not generated in perfect alignment to the JEDEC DDR specifications (see Figure 1).¹

In the timing values supplied, the timings between MemClk0, data bits, and strobes are skewed from the ideal case. The primary reasons behind the signal skew are: signal routing internal to the chip and package; delays through the output buffers; and loading on the external interface lines.

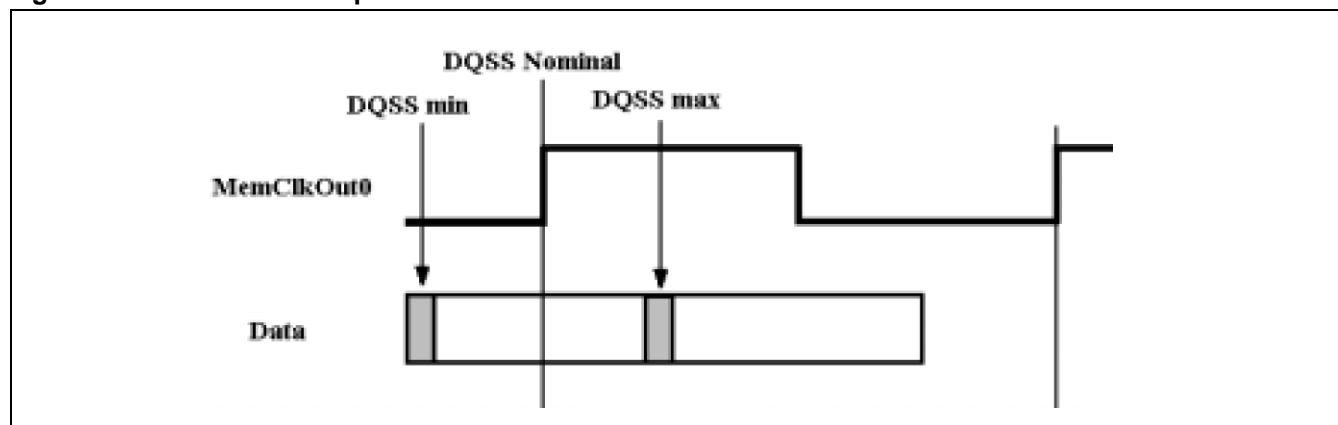
Most of this skew can be corrected by advancing MemClkOut0 in relation to the data bits and associated strobe line. Basically, the data bit and strobe signals timings are not changed, but MemClkOut0 is advanced by 90° (one quarter-cycle). This more closely aligns MemClkOut0 to the data and data strobe signals, thereby more closely approaching the ideal case. This approximate realignment enables most small point-to-point memory systems (three chips or less) and SO-DIMM-based or DIMM-based memory systems to perform memory writes properly.

DDR memory devices systems have a relatively wide data acceptance window when data is being written to them (see Figure 2). Basically, this means that data and its strobe signal can arrive approximately one quarter-clock cycle before and up to one quarter-clock cycle after the edge transition of MemClkOut0.

1. The PowerPC 440GP Data Sheet lists the worst-case delays associated with the signal groups in question. A main design component missing from the PowerPC 440GP Data Sheet is the delay associated with the output buffers driving the physical signal lines. This delay is set by the physical design of the printed circuit board and memory configuration. When performing the needed level of timing analysis for a DDR memory system, an IBIS or encrypted h-Spice model of the output buffers is needed along with the specifics of the eventual board design.

Note: The real usable time will be smaller than the half-clock cycle window due to jitter, skew, and other factors that affect the signals. Barring poor design and other factors, the majority of memory writes will work after the memory clock is advanced by 90°.

Figure 2: Write Data Acceptance Window



Data Read Operations

The primary use of the digital delay line is to aid in capturing data read from the memory system. The delay line is used when adjusting the timing of the data capture logic contained in the read data path. Data arriving from the memory system is delayed from the internal time domain of the microprocessor. The delay is associated with the signal path lengths, loading, and delays internal to the memory chips. Before the arriving read data can be captured and transferred to its eventual destination, it must be time-aligned with the internal operation of the processor chip. The read data path performs the capture and realignment of the read data. See Figure 3 when reviewing the operation of the read data path.

Read Data Path

When data is launched from the memory system, the leading edge of data is edge-aligned with the leading edge of the associated byte lane strobe signals. With a properly designed printed circuit board, the strobe and data signals arrive at the input pins of the PowerPC 440GP/GX in this same alignment. The read data path is used to realign the incoming data and its external memory system time domain with the internal time domain of the PowerPC 440GP/GX chip. This realignment is performed using one of the following two methods:

- If the system delays and path lengths are short enough (this is rarely the case), then data can be routed almost directly to the PLB bus and on to the requesting internal resources.
- The data is delayed by a certain time value in relation to the PLB time domain so that data can be reliably captured. This is the usual method of realigning the data. The delay amount is determined by the accumulated delays associated with the read data and the adjustments made to the DDR timing registers.

The read data path is composed of the following three stages:

- An initial data capture stage (Stage 1).
- A second delay able/realignment stage (Stage 2).
- A third and final stage is primarily used in conjunction with CAS 2.5 devices, ECC memory systems, and larger DIMM memory systems (Stage 3).

The final stage (Stage 3) in the read data path is the PLB read sample point. Stage 3 can be used to send the proper memory read data to the PLB bus and then to the requesting master device. The read sample point is used to pick the proper time or step to transfer the incoming data to the PLB.

Note: Each stage used in the read data path adds one PLB clock cycle of latency to the data.

Stage 1

Stage 1 captures the read data arriving from the memory system. Data arriving from the memory system should arrive with data strobes edge-aligned with the leading edge of data. Data strobes are delayed by one quarter-clock cycle when entering the read data path. (In essence, the strobe is now center-aligned with received data.) This enables the data to have the proper setup and hold time for the input flip-flops (FF) and transparent latch (XL).

Stage 1 captures and transfers data arriving in two discrete times: on the rising edge and then on the falling edge of MemClkOut0. Because data is delivered from the memory system in two units per clock cycle, there needs to be a method of gathering up the data and transferring it in to the PLB in a single clock cycle. The read data path is therefore split into two paths: the upper data path and the lower data path.

The upper data path latches the data that arrive first on the rising edge of MemClkOut0. The lower data path uses a transparent latch to acquire the second unit of data associated with this memory clock cycle. The first arriving data unit is latched with the delayed data strobe and held on the outputs until the next data strobe arrives. The second arriving unit of data (the falling edge of MemClkOut0) is flushed through the lower data path while the latch is in transparent mode. Data is captured on the falling edge of MemClkOut0 and held for less than one half-clock cycle, until MemClkOut0 rises, forcing the latch to become transparent again. By the end of this process, both units of data have been captured in Stage 1 and are ready to be passed on to Stage 2 of the read data path.

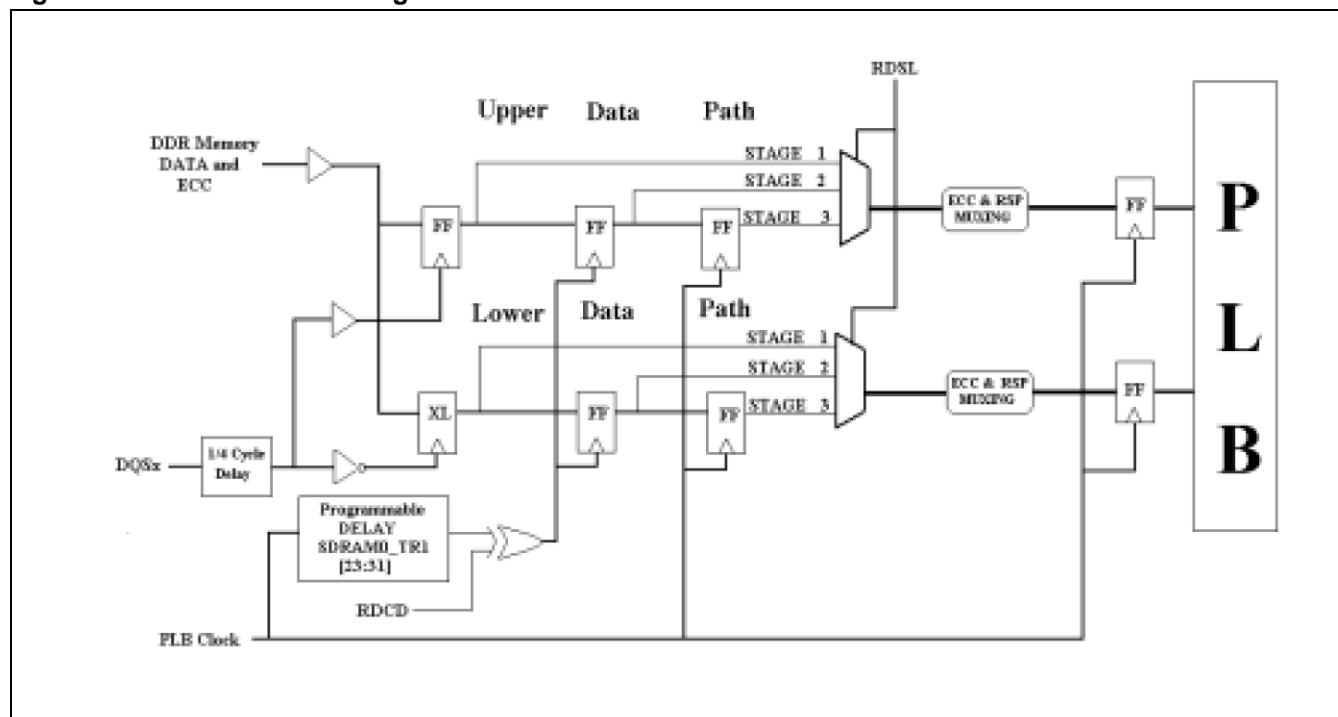
Note: When using Stage 1 to correct for byte lane skew, the timing of Stage 2 is reduced. The remaining time available to pass data to Stage 2 must be verified with a detailed timing budget. If proper design rules have been followed, there should be minimal skew between the byte lanes and strobe signals. After data is captured in Stage 1, both units of data (the rising edge and the falling edge of MemClkOut0) are ready to be transferred simultaneously to the next stage of the read data path.

Stage 2

Stage 2 of the read data path realigns the incoming data from Stage 1 and its memory system time domain to the internal PLB time domain of the processor. This stage can delay the incoming data by up to a half-clock cycle in increments of full and one quarter-delay elements. See Figure 3.

The key for proper Stage 2 tuning is the development of a detailed timing budget for the read data path. The timing budget defines when data will arrive at the input pins of the processor and therefore predicts the time alignment of memory data to the internal PLB time domain. The timing budget must contain all of the delay items associated with the physical path (transit times), memory parts, and on-chip delays.

Figure 3: Read Data Path diagram



The PLB expects to transfer data on the rising edge of its clock. The programmable delay associated with Stage 2 is used to place data on the appropriate side of the PLB clock. If data arrives from Stage 1 of the read data path near the rising edge of the PLB clock and does not have enough time to meet setup and hold times of the flip-flops of Stage 2, then data must be delayed until after the rising edge of the PLB clock. If data arrives just after a rising edge of PLB clock, then nothing should need to be done. It is important to determine whether data will have enough time to be latched properly by Stage 2. In other words, will data appear at the outputs of Stage 2 in time to be passed on to the PLB?

Note: Another constraint affecting the programmable delay value on Stage 2 is the Error Checking and Correcting (ECC) function. Because the ECC function takes time to perform, the timing values will probably need to be modified if ECC is enabled. This is described in Effects of the ECC Function on page 7 and in additional application notes.

Bits 23:31 of the SDRAM0_TR1 register are used when setting the programmable delay time of Stage 2. The digital delay line calibration value is the basis of the time calculations. The programmable delay attached to this portion of the data path is used to vary the point in time at which the data is latched by and passed on to one of the following:

- Stage 3 in the read data path
- ECC checking portion of the path
- Final latches where data is passed on to the PLB

Stage 3

Stage 3 consists of a final set of flip-flops clocked by the PLB clock. The flip-flops take data from Stage 2 and present it either to the PLB (where it will be passed on the requesting processor resource) or to the ECC logic. It takes 4.0 ns to check the data for single-bit and double-bit errors. When captured in Stage 3, data will have nearly a full memory clock cycle to perform the ECC operation or progress on to the PLB. Stage 3 timing cannot be adjusted or manipulated. One additional clock period of latency is the penalty for using this stage of the read data path.

Effects of the ECC Function

When the PowerPC 440GP/GX uses the Error Checking and Correcting (ECC) function, the memory systems timings and the settings of the read data path usually have to be changed. The reason for this is that it takes approximately 4.0 ns to perform the ECC process on read data. (ECC adds or enables an extra data byte and data strobe on the PowerPC 440GP/GX DDR interface.) The ECC value is calculated before data is written to the memory system and is presented simultaneously with data to the memory system.

Note: ECC data writes are transparent to the data write process and appear to be the same as standard non-ECC writes. From the write point-of-view, therefore, the ECC bits are just part of the standard write data package and there is nothing special about the added ECC bits of data or timing of the memory interface.

Because it takes approximately 4.0 ns to perform the ECC function, data needs to appear at the beginning of the ECC functional block at least 4.0 ns early (plus the setup time for the final stage of flip-flops, plus error/jitter factors). All of the flip-flops used in the DDR section of the PowerPC 440GP/GX have a 200 ps setup time and 100 ps hold time. Because of the additional delays, data usually will have to flow through all three stages of the read data path when using the ECC function.

For the vast majority of memory systems, the system timing budget does not allow the memory system to use ECC and still operate with the same timings as non-ECC memory systems. Usually, there is not enough time to perform the ECC function for read data path stage timings. If there is enough time remaining in a PLB clock cycle, data can be checked while data is in Stage 2. Usually, however, read data will need to be passed on to Stage 3 of the read data path after realigning the data's system timing in Stage 2. After it is passed on to Stage 3, data has nearly one full clock cycle to pass through the ECC function and then on to the PLB. The main drawback of this additional stage is one additional latency cycle penalty.

Read Data Path Control

Several control signals shown in Figure 3 are not described in detail. These additional signals are used to select the paths and stages data will follow through the read data path and to define when the data is loaded on to the PLB. The signals are described in detail in the PowerPC 440GP User's Manual and the PowerPC 440GX User's Manual.

As shown in Figure 3, data can pass through up to three stages to reach the PLB. Each stage takes a different amount of time to reach the PLB. The RDSS setting determines when the read data is to be loaded on to the PLB from the read data path. The PowerPC 440GP User's Manual and the PowerPC 440GX User's Manual explain why various sample stages are chosen.

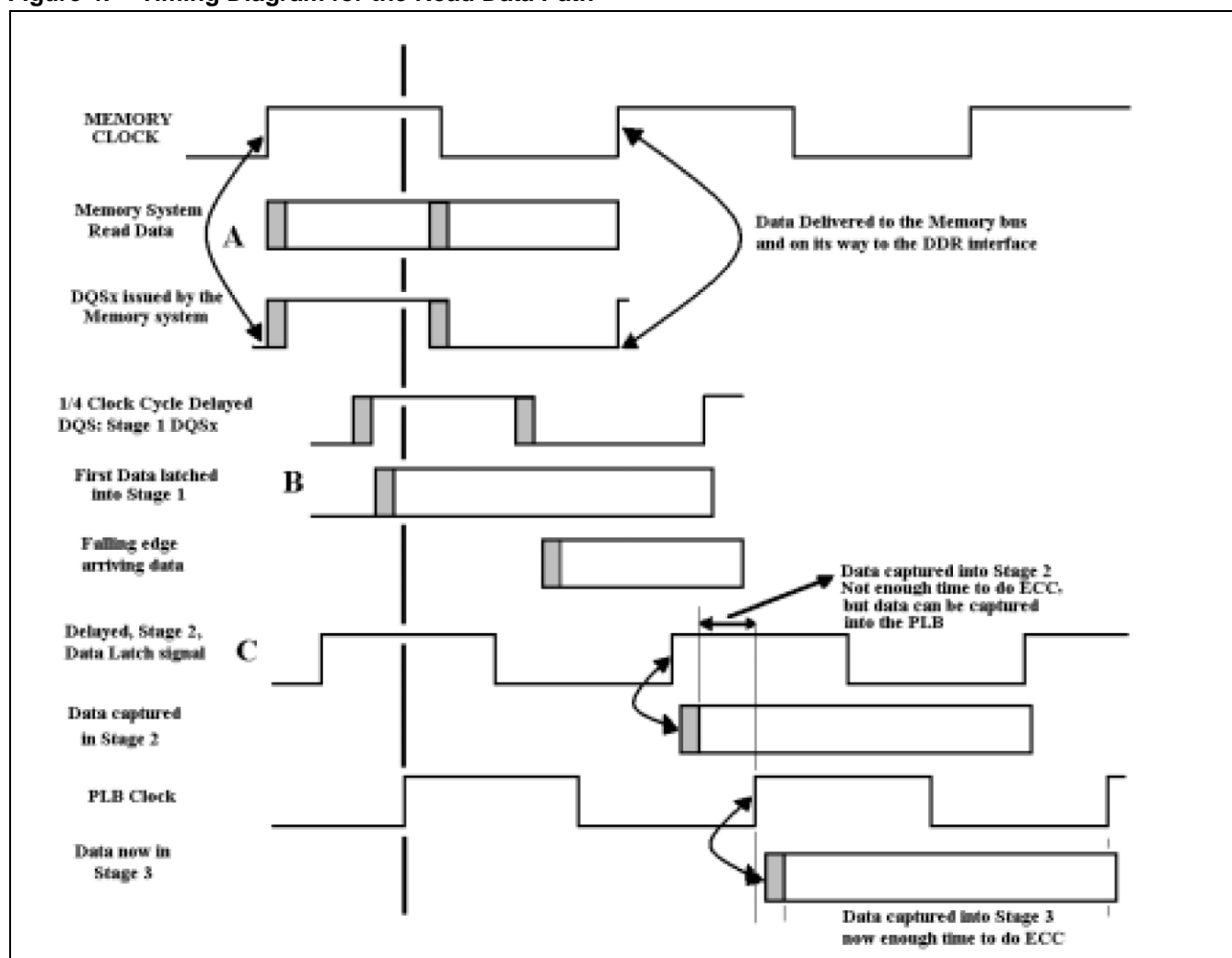
The RDCD control line is used to insert a half-cycle delay into Stage 2 of the read data path. This setting is used for CAS 2.5 memory devices. The setting is used when realigning CAS 2.5 data where the first unit of data is delivered on the falling edge of MemClkOut0 (the fifth clock edge after the read command is delivered). It enables read data to be captured into Stage 3 while compensating for the half-cycle latency and allowing proper synchronization with the internal DDR SDRAM clock domain.

Note: CAS 2.0 data is delivered on the rising edge of MemClkOut0 (the fourth clock edge after the command is delivered).

The RDCD bits contain the delay setting for read data path Stage 2 settings. The bits used in this register, SDRAM0_TR1[23:31], set the delay amount used by Stage 2. The register is divided into two zones. Bits 23:29 are used to hold full digital delay line element values. Bits 30:31 are used to contain quarter-delay element values. Generally, the quarter-delay element bits are not needed when setting the delay time.

Note: The total delay programmed into the bit field should not exceed a half-clock cycle.

Figure 4: Timing Diagram for the Read Data Path



Memory System Example

Consider the following example, shown in Figure 4, in which MemClkOut0 is advanced by 90°.

Data is delivered from the memory system and its time domain to Stage 1 of the read data path (see A in Figure 4). It is important to calculate both when the data is delivered to Stage 1 and what relationship the data has to the PLB clock signal. Use these calculations to establish the point at which data is delivered in relation to the PLB clock cycle. After this point is found, start working the signal through the read data path.

When data arrives at Stage 1 of the read data path, both units of data (the rising and falling edge) must be captured and presented simultaneously to Stage.

2. The first appearing data (the rising edge of MemClkOut0) is delayed by one quarter-clock cycle in Stage 1 (see B in Figure 4) and then captured and appears on the output of the flip-flops. The second unit of data (the falling edge of MemClkOut0) is captured after the delay period and is held on the outputs of the transparent latch for nearly a half-clock cycle. The point in time at which the data is valid on the output side of Stage 1 determines the programmable delay time to be used with Stage 2 of the read data path.

The objective of Stage 2 timing is to acquire both units of data, rising and falling edge, and to move them on to the PLB (or the input of the ECC logic). If data is presented to the inputs of Stage 2 near the rising edge of PLB clock (within the setup, plus jitter, plus a small data valid window, 300-400 ps), then data should be delayed until after the rising edge of PLB clock. If there is adequate setup time to capture data, it might not need to be delayed, but it might be latched directly into Stage 2 and on to the PLB.

For this example, data needs to be delayed to appear beyond the rising edge of the PLB clock (see C in Figure 4). The delay amount is derived from the digital delay line calibration value and the time delay quantity. The delay value is in units of "delay elements". The time per delay element is derived from the operational frequency of the DDR interface, MemClkOut0, and the count of delay elements in a half-clock cycle from the SDRAM0_DLYCAL register. Convert the desired Stage 2 delay time into a delay element count and install the value in the SDRAM0_TR1[RDCT] register. Data from Stage 1 will now be delayed and appear beyond the rising edge of the PLB clock.

In this particular case, there is enough time to perform the ECC function. The data has arrived at the outputs of Stage 2 very near the beginning of a PLB clock cycle. In this case there is nearly a full clock cycle available to perform the ECC function (a 4.0 ns process) and then to have the data placed on the PLB and sent to the requesting party.

In a case where data is captured in Stage 2 near the end of a PLB clock cycle, data would need to be passed on to Stage 3. When data is passed on to Stage 3, there will be nearly a full PLB clock cycle to perform the ECC function and then to pass the results on to the PLB. This additional stage adds one additional clock cycle of latency to the read data.

Final Points

There are several additional issues to keep in mind when configuring and using the DDR SDRAM interface.

- After the DDR interface is configured and operating, the timing values cannot be altered. If the timing values are altered while the interface is operating, there is no guarantee that the memory system will be able to track the changes and continue to deliver valid data.
- When considering a small point-to-point memory system (three memory devices or less), minimize the signal line lengths to make them as short as possible, but match the signal lengths to minimize signal skew.
- The DDR SDRAM interface can be configured in several ways and still work properly. It is the responsibility of the end user to optimize the interface timing and read data path stage selection in order to minimize data latency and assure correct data capture.

One last point to address is the manipulation of various clock delay tuning bits. Whereas some bits in the various control registers advance the main memory clock (MemClkOut0), some of the bits incrementally delay the advanced clock. Generally, MemClkOut0 is advanced in approximate increments of 90°. The delay function incrementally backs up the advanced clock, so that:

total clock advance = advance value - delay values

This provides the end user the ability to fine-tune the total advance on the MemClkOut0 line and the appearance of read data to the DDR interface on the processor.

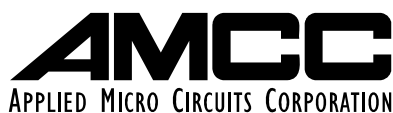
Refer to the PowerPC 440GP User's Manual or the PowerPC 440GX User's Manual for detailed timing information.

Conclusions

The flexibility found in the PowerPC 440GP/GX can lead to incorrect configurations of the interface. By explaining in detail the digital delay line and read data path configuration and tuning, this application note seeks to clarify some of the more difficult concepts of using the DDR interface. If the end user understands the basics of the DDR SDRAM interface, the tuning process should be greatly simplified. With the flexibility found in the DDR controller, settings can be configured in several different ways to achieve the desired results. After the basic components of the interface are understood, tuning in the physical memory system should proceed more rapidly.

Document Revision History

Revision	Date	Description
v1.01	1/18/08	Converted layout to AMCC format.



Applied Micro Circuits Corporation
6310 Sequence Dr., San Diego, CA 92121

Main Phone: (858) 450-9333 — Technical Support Phone: (858) 535-6517 — (800) 840-6055
<http://www.amcc.com> (support@amcc.com)

AMCC reserves the right to make changes to its products, its datasheets, or related documentation, without notice and warrants its products solely pursuant to its terms and conditions of sale, only to substantially comply with the latest available datasheet. Please consult AMCC's Term and Conditions of Sale for its warranties and other terms, conditions and limitations. AMCC may discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information is current. AMCC does not assume any liability arising out of the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others. AMCC reserves the right to ship devices of higher grade in place of those of lower grade.

AMCC SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

AMCC is a registered Trademark of Applied Micro Circuits Corporation. Copyright © 2008 Applied Micro Circuits Corporation.

I2C BUS® is a registered Trademark of Philips N.V. Corporation Netherlands.